Week 8 - Exercise 13

Binay Prasanna Jena

October 24th 2020

Fit a Logistic Regression Model to the Thoracic Surgery Binary Dataset

Problem Statement : For this problem, you will be working with the thoracic surgery data set from the University of California Irvine machine learning repository. This dataset contains information on life expectancy in lung cancer patients after surgery.

The underlying thoracic surgery data is in ARFF format. This is a text-based format with information on each of the attributes. You can load this data using a package such as foreign or by cutting and pasting the data section into a CSV file.

```
## Set the working directory to the root of your DSC 520 directory
setwd("C:/Users/binay/Documents/GitHub/dsc520/")

## Load the 'foreign' library
library(foreign)

## Load the 'data/ThoracicSurgery.arff' to
thoracic_surgery_df <- read.arff('data/ThoracicSurgery.arff')
head(thoracic_surgery_df)
```

```
##    DGN PRE4 PRE5 PRE6 PRE7 PRE8 PRE9 PRE10 PRE11 PRE14 PRE17 PRE19 PRE25 PRE30
## 1 DGN2 2.88 2.16 PRZ1    F    F    F     T     T  OC14     F     F     F     T
## 2 DGN3 3.40 1.88 PRZ0    F    F    F     F     F  OC12     F     F     F     T
## 3 DGN3 2.76 2.08 PRZ1    F    F    F     T     F  OC11     F     F     F     T
## 4 DGN3 3.68 3.04 PRZ0    F    F    F     F     F  OC11     F     F     F     F
## 5 DGN3 2.44 0.96 PRZ2    F    T    F     T     T  OC11     F     F     F     T
## 6 DGN3 2.48 1.88 PRZ1    F    F    F     T     F  OC11     F     F     F     F
##   PRE32 AGE Risk1Yr
## 1     F  60       F
## 2     F  51       F
## 3     F  59       F
## 4     F  54       F
## 5     F  73       T
## 6     F  51       F
```

Attribute Information:

1. DGN: Diagnosis - specific combination of ICD-10 codes for primary and secondary as well multiple tumours if any (DGN3,DGN2,DGN4,DGN6,DGN5,DGN8,DGN1)

2. PRE4: Forced vital capacity - FVC (numeric)
3. PRE5: Volume that has been exhaled at the end of the first second of forced expiration - FEV1 (numeric)
4. PRE6: Performance status - Zubrod scale (PRZ2,PRZ1,PRZ0)
5. PRE7: Pain before surgery (T,F)
6. PRE8: Haemoptysis before surgery (T,F)
7. PRE9: Dyspnoea before surgery (T,F)
8. PRE10: Cough before surgery (T,F)
9. PRE11: Weakness before surgery (T,F)
10. PRE14: T in clinical TNM - size of the original tumour, from OC11 (smallest) to OC14 (largest) (OC11,OC14,OC12,OC13)
11. PRE17: Type 2 DM - diabetes mellitus (T,F)
12. PRE19: MI up to 6 months (T,F)
13. PRE25: PAD - peripheral arterial diseases (T,F)
14. PRE30: Smoking (T,F)
15. PRE32: Asthma (T,F)
16. AGE: Age at surgery (numeric)
17. Risk1Y: 1 year survival period - (T)rue value if died (T,F)

a. Fit a binary logistic regression model to the data set that predicts whether or not the patient survived for one year (the Risk1Y variable) after the surgery. Use the glm() function to perform the logistic regression. See Generalized Linear Models for an example. Include a summary using the summary() function in your results.

```
# Since Risk1Y = True means patient died and we want to predict whether patient survived.
# We should set baseline category as True, (which means Not Survived)
# otherwise default will be taken as False due to alphabetical order
# This means T = 0 and F = 1
thoracic_surgery_df$Risk1Yr<-relevel(thoracic_surgery_df$Risk1Yr, "T")

# Could also split this df using split function to training set and test set.
# However leaving it for this exercise, assuming all dataset is training dataset.

# This model includes all other parameters as dependent
lrmodel.1 <- glm(Risk1Yr ~ . , family ='binomial' , data = thoracic_surgery_df)
summary(lrmodel.1)
```

```
##
## Call:
## glm(formula = Risk1Yr ~ ., family = "binomial", data = thoracic_surgery_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4929   0.2762   0.4199   0.5439   1.6084
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.655e+01  2.400e+03   0.007  0.99450
## DGNDGN2     -1.474e+01  2.400e+03  -0.006  0.99510
## DGNDGN3     -1.418e+01  2.400e+03  -0.006  0.99528
```

```
## DGNDGN4     -1.461e+01  2.400e+03  -0.006  0.99514
## DGNDGN5     -1.638e+01  2.400e+03  -0.007  0.99455
## DGNDGN6     -4.089e-01  2.673e+03   0.000  0.99988
## DGNDGN8     -1.803e+01  2.400e+03  -0.008  0.99400
## PRE4         2.272e-01  1.849e-01   1.229  0.21909
## PRE5         3.030e-02  1.786e-02   1.697  0.08971 .
## PRE6PRZ1     4.427e-01  5.199e-01   0.852  0.39448
## PRE6PRZ2     2.937e-01  7.907e-01   0.371  0.71030
## PRE7T       -7.153e-01  5.556e-01  -1.288  0.19788
## PRE8T       -1.743e-01  3.892e-01  -0.448  0.65419
## PRE9T       -1.368e+00  4.868e-01  -2.811  0.00494 **
## PRE10T      -5.770e-01  4.826e-01  -1.196  0.23185
## PRE11T      -5.162e-01  3.965e-01  -1.302  0.19295
## PRE14OC12   -4.394e-01  3.301e-01  -1.331  0.18318
## PRE14OC13   -1.179e+00  6.165e-01  -1.913  0.05580 .
## PRE14OC14   -1.653e+00  6.094e-01  -2.713  0.00668 **
## PRE17T      -9.266e-01  4.445e-01  -2.085  0.03709 *
## PRE19T       1.466e+01  1.654e+03   0.009  0.99293
## PRE25T       9.789e-02  1.003e+00   0.098  0.92227
## PRE30T      -1.084e+00  4.990e-01  -2.172  0.02984 *
## PRE32T       1.398e+01  1.645e+03   0.008  0.99322
## AGE          9.506e-03  1.810e-02   0.525  0.59944
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 395.61  on 469  degrees of freedom
## Residual deviance: 341.19  on 445  degrees of freedom
## AIC: 391.19
##
## Number of Fisher Scoring iterations: 15
```

```
# Trying out another model
# This model includes only DGN parameters as dependent
lrmodel.2 <- glm(Risk1Yr ~ DGN , family ='binomial' , data = thoracic_surgery_df)

summary(lrmodel.2)
```

```
##
## Call:
## glm(formula = Risk1Yr ~ DGN, family = "binomial", data = thoracic_surgery_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0464   0.5128   0.5128   0.5128   1.1774
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.557e+01  1.455e+03   0.011    0.991
## DGNDGN2     -1.436e+01  1.455e+03  -0.010    0.992
## DGNDGN3     -1.360e+01  1.455e+03  -0.009    0.993
## DGNDGN4     -1.382e+01  1.455e+03  -0.009    0.992
## DGNDGN5     -1.543e+01  1.455e+03  -0.011    0.992
```

```
## DGNDGN6      1.300e-08  1.627e+03   0.000    1.000
## DGNDGN8     -1.557e+01  1.455e+03  -0.011    0.991
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 395.61  on 469  degrees of freedom
## Residual deviance: 379.79  on 463  degrees of freedom
## AIC: 393.79
##
## Number of Fisher Scoring iterations: 14
```

b. According to the summary, which variables had the greatest effect on the survival rate?

Based on the summary, below values have significant effect, as mentioned in order of significance. So the variables which have greatest effect are in their order of effect

1. PRE9
2. PRE14
3. PRE30
4. PRE17
5. PRE5

Details from Summary Above:

PRE9T -1.368e+00 4.868e-01 -2.811 0.00494 **

PRE14OC14 -1.653e+00 6.094e-01 -2.713 0.00668 **

PRE30T -1.084e+00 4.990e-01 -2.172 0.02984 *

PRE17T -9.266e-01 4.445e-01 -2.085 0.03709 *

PRE14OC13 -1.179e+00 6.165e-01 -1.913 0.05580 .

PRE5 3.030e-02 1.786e-02 1.697 0.08971 .

Signif. codes: 0 ‘’ ***0.001*** ‘’ *0.01* ‘’ 0.05 ‘.’ 0.1 ‘ ’ 1

c. To compute the accuracy of your model, use the dataset to predict the outcome variable. The percent of correct predictions is the accuracy of your model. What is the accuracy of your model?

```
result <- predict(lrmodel.1,thoracic_surgery_df,type = "response")
# result
# validating - putting the actual value and counts of Predicted values in a matrix
# Since T = 0 and F = 1, so if result < 0.5, it should be T
confmatrix <- table(ActualValue=thoracic_surgery_df$Risk1Yr, PredictedValue = result < 0.5)
confmatrix
```

```
##             PredictedValue
## ActualValue FALSE TRUE
##           T    67    3
##           F   390   10
```

```r
# accuracy - Cases where we predicted correctly by Total Predictions
# from matrix, we see when Actual Value is T, confmatrix needs to pick 1,2
# and when F it should pick 2,1
(confmatrix[1,2]+confmatrix[2,1])/sum(confmatrix)
```

```
## [1] 0.8361702
```

this model shows an accuracy of ~83% approximately. Also if we look the matrix, we see that this model is better predicting a F i.e. if a person lives or survives rather than predicting whether the person would die within the year.